(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 03152624 A

(43) Date of publication of application: 28 . 06 . 91

(51) Int. Cl G06F 7/50
G06F 9/34

(21) Application number: 01291799

(22) Date of filing: 09 . 11 . 89

(71) Applicant: RICOH CO LTD

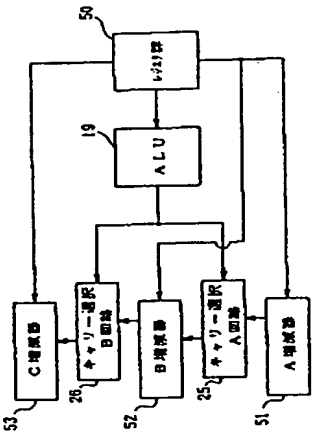
(72) Inventor: YASUI TAKASHI
YOSHIOKA KEIICHI
YAMAURA SHINICHI

(54) CENTRAL ARITHMETIC PROCESSOR

(57) Abstract:

PURPOSE: To shorten the processing cycle of a central arithmetic processor without loosing the parallel arithmetic performance by adding a carry signal produced from an arithmetic process of a data arithmetic unit to the number of adders to be added via an increment device in order to obtain the address data.

CONSTITUTION: In the register/indirect 8-bit displacement, an arithmetic and logic unit ALU 19 calculates the 8-bit displacement and the lower 8 bits of the data on a register W2. Then the carry signal obtained from the calculation is selected by a carry selection circuit 25. The circuit 25 sends the carry signal to a B increment/decrement device 52. On the other hand, the high-order data, etc., are transferred to the device 52 and a C increment/decrement device 53. Then the A-C increment devices 51-53 work after deciding the increment or the decrement of a constant 0 based on the code extension value. In a register indirect 16-bit displacement, the arithmetic result is supplied to a carry selection B circuit 26 and then sent to the device 53. Thus the addresses are calculated by the ALU 19 and the devices 51-53. Then the processing cycle is shortened while maintaining the parallel arithmetic performance.



⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　平3－152624

⑤Int. Cl.⁵	識別記号	庁内整理番号	⑬公開　平成3年(1991)6月28日
G 06 F 7/50	E	7056－5B	
	Z	7056－5B	
9/34			
		7927－5B　G 06 F 9/36　310	

審査請求　未請求　請求項の数 1　（全12頁）

⑤④発明の名称　中央演算処理装置

⑳特　願　平1－291799
㉒出　願　平1(1989)11月9日

⑦②発明者　安井　隆　東京都大田区中馬込1丁目3番6号　株式会社リコー内
⑦②発明者　吉岡　圭一　東京都大田区中馬込1丁目3番6号　株式会社リコー内
⑦②発明者　山浦　慎一　東京都大田区中馬込1丁目3番6号　株式会社リコー内
⑦①出願人　株式会社リコー　東京都大田区中馬込1丁目3番6号
⑦④代理人　弁理士　青山　葆　外1名

明　細　書

1．発明の名称

中央演算処理装置

2．特許請求の範囲

（1）アドレスのビット数を複数に分割したビット数を処理し、供給される複数の定数のいずれか一つを被増加数に加算しアドレスデータを作成する複数の増加器と、

少なくとも一つのレジスタと、

上記レジスタが送出する、アドレスのビット数より少ないビット数からなるデータに基づき演算を行う一つのＡＬＵと、

上記ＡＬＵの演算結果で桁上げ信号が発生した場合にはこの桁上げ信号を上記増加器に送出するキャリー信号選択手段と、を備えたことを特徴とする中央演算処理装置。

3．発明の詳細な説明

〔産業上の利用分野〕

本発明は、中央演算処理装置に関する。

〔従来の技術とその課題〕

中央演算処理装置においてデータの演算処理を実行する際に必要なアドレス信号を発生する方法として、データ演算用のＡＬＵとは独立して設けられるフルアダーにて構成されアドレス信号のみを発生するアドレス計算ユニットを設ける方法と、データの演算とアドレス信号の発生との両方をともに行う一つのＡＬＵを設ける方法とがある。

前者のアドレス計算ユニットを設ける方法においては、ＡＬＵとアドレス計算ユニットとを別々に設けなければならず、ハードウエアが大きくなるという問題点がある。一方、後者の一つのＡＬＵを設ける方法においては、データ演算とアドレス計算とを行うためそれぞれの演算を別々に行うことができず並列性がなくなり、演算に要する処理サイクルが多くなるという問題点がある。

本発明はこのような問題点を解決するためになされたもので、ハードウエアが大きくならず、かつ演算の並列性を損なわない中央演算処理装置を提供することを目的とする。

〔課題を解決するための手段〕

本発明は、アドレスのビット数を複数に分割したビット数を処理し、供給される複数の定数のいずれか一つを被増加数に加算しアドレスデータを作成する複数の増加器と、

少なくとも一つのレジスタと、

上記レジスタが送出する、アドレスのビット数より少ないビット数からなるデータに基づき演算を行う一つのＡＬＵと、

上記ＡＬＵの演算結果で桁上げ信号が発生した場合にはこの桁上げ信号を上記増加器に送出するキャリー信号選択手段と、を備えたことを特徴とする。

〔作用〕

キャリー信号選択手段は、ＡＬＵにおける演算処理にてキャリー信号が発生した場合にはこのキャリー信号を増加器に送出し、増加器は被加算数にキャリーデータを加算しアドレスデータを発生する。このようにＡＬＵと増加器とによりアドレスデータを発生することは、ＡＬＵ単体でアドレスデータを発生する場合に比べ並列性を失わず処理サイクルを短くするように作用し、又、アドレスデータ計算用のＡＬＵを有する必要がないことよりハードウエアが小さくなるように作用している。

〔実施例〕

まず、本発明の中央演算処理装置における一実施例における構成の概略を第３図ないし第５図a,b,cを参照し以下に説明する。

第４図は、プログラミングモデルであり、本中央演算処理装置(以下ＣＰＵと称す)の基本語長は８ビットである。

アドレス空間は、プログラムをアクセスする際、プログラムカウンタ(以下ＰＣと記す)は２４ビット(ＰＢＣ、ＰＣＨ、ＰＣＬ)を有しており、リニアアドレスで１６Ｍバイトをアクセス可能としている。尚、ＰＢＣとはプログラム・バンク・カウンタレジスタ(以下ＰＢＣと記す)、ＰＣＨとはプログラム・カウンタレジスタＨ(以下ＰＣＨと記す)、ＰＣＬとはプログラム・カウンタレジスタＬ(以下ＰＣＬと記す)である。

一方、データをアクセスする時は、バンク方式を用いており、バンクアドレスとしては、基本的にデータバンクレジスタ(以下ＤＢＲと記す(８ビット))が出力され、従って、６４Ｋバイトリニアで２５６バンクを用いて、１６Ｍバイトのアクセスを可能にしている。

尚、バンクアドレスとしてのＤＢＲの出力については後述のＭ１,Ｍ０フラグにて説明する。

また、複数の汎用レジスタ(Ｗ０～Ｗ３:１６ビット)があり、特に、Ｗ０,Ｗ１の両レジスタは８ビットごとに分別され、８ビットレジスタＲ０,Ｒ１,Ｒ２,Ｒ３として使用することもできる。

故に、本ＣＰＵでは、演算のデータサイズとして８ビット,１６ビットの両方のサイズのデータを命令により区別して扱うことが可能である。

さらに、スタック空間としては、スタックポインタレジスタ(以下ＳＰと記す)として１６ビットレジスタを用意しており、リニアに６４Ｋバイトのアクセスをおこなう。ただし、バンクアドレスは、"００"hに固定されている。

そして、プログラムステイタスレジスタ(以下ＰＳＲと記す)は現在のＣＰＵの動作状態を示しており、具体的には、Ｎ,Ｖ,Ｚ,Ｃの各フラグは、演算の結果により変化し、Ｉフラグは、割り込み要求の受付けの可否を示し、Ｄフラグは、加減算命令の結果の補正に関し、Ｄ＝１ならば、加減算命令の実行結果は自動的に１０進補正される。

Ｍ１,Ｍ０フラグは、データ空間をアクセスする際、出力されるバンクアドレスの選択を可能にするフラグである。従って、Ｍ１,Ｍ０フラグを任意の値に設定(このＣＰＵでは命令で更新する)する事により、データアクセスの際に、出力されるバンクアドレスをＤＢＲ値、"００"h等の定数の何れかを選択して出力し様々なメモリのアプリケーションに対応させる。

ファーストページレジスタ(以下ＦＰＲと記す)はデータアクセス時のアドレス・ポインタとなるレジスタで、ファースト・ダイレクトと呼ぶアドレッシング・モードで使用される。尚、アドレッシングとは、データの格納先のアドレスを指定することをいう。

このアドレッシング・モードではオペランド・データとして8ビットのデータのみをフェッチしてそのデータを実効アドレスのロー(ビット7～ビット0)とし、ハイ(ビット15～ビット8)をFPRの内容とするアドレッシングモードにおいて有効となるレジスタである。

ただしこのときも、出力されるバンクアドレスは、M1,M0のフラグ状態に従う。

このアドレッシング・モードは、オペランドデータを1バイトのみフェッチするだけなので、同一ページアドレス内(アドレスのビット15～ビット8が一定値)の高速なデータのアクセスが可能となる。

第5図aないし第5図cは、本CPUの命令形式について示したものであり、このCPUは基本語長は、前述のように8ビットであり、オペコードの前にプリバイトと呼ばれる命令拡張用の1バイトデータをフェッチする形式をとる。

基本的にプリバイト・データは、アドレッシング・モードに係る情報を有し、オペコードが実行すべき命令の内容を持っている。

但し、命令の使用頻度が高いものについては、命令コード長と実行時間の短縮を図るため、「形式1」に示すように、短縮命令と呼ぶプリバイトの無いオペコード内にアドレッシング及び命令の内容を含んだ命令を用意する。

さらに、オペランドデータは2種類の配置形式をもっている。第5図bに示す「形式2」は、プリバイトの次にオペコードを配置し、その後にオペランドデータを配する形式であり、「形式3」はプリバイトとオペコードの間にもオペランドデータを配置する。

特に、形式3のプリバイトとオペコードの間のオペランドデータは、ディスプレースメント付アドレッシングで使用される。

ここでいうディスプレースメント付アドレッシングとは、データのアクセスのための実効アドレスの発生時に、内部レジスタデータにオペランドでフェッチされたデータもしくは、オペランドで指定されたレジスタの値をオフセットとして加算することで実効アドレスを発生するアドレッシングのことを示す。

このアドレッシング・モードが使用される際、もし形式2のようなオペランドデータの配置形式を取れば、ディスプレースメントのオペランドデータをフェッチした後、実効アドレスを計算するために、時間を要し、オペランドのディスプレースメント・データのフェッチの後、複数のアイドルサイクルが存在することになる。

しかし、形式3の配置をこのとき用いて、プリバイトとオペコードの間にディスプレースメントデータを配置すれば、実効アドレス発生のための計算をオペコードのフェッチサイクルに重複して行なう事ができ、無駄なアイドルサイクルの発生を防ぐ。

第3図は、ブロックレベルの構成図であり、本CPUは主に制御部1と演算部2の2つの機能部に大別される。

初めに、制御部1であるがここは、命令の実行を制御する機能を持っている。

動作としては、命令の実行に際し、外部からデータバス(D7～D0)を介して、DIL15に入力された命令コードは、プリバイトIR3或いはオペコードIR4の各インストラクションレジスタに格納され次の命令が発生するまで保持される。

そして、これらのインストラクションレジスタの複数の出力5,6と命令シーケンスのタイミングを制御するTCU7の出力がAND－ORのPLAで構成された命令デコード回路8,9,10,11に入力され、命令とタイミングに応じたデコード結果13を出力する。

さらにそのデコード結果は、EC112というインターフェース回路を介して、演算部2に対してタイミングを整えて演算部2を制御すべき複数の制御信号14を発生する。

但し、本CPUにおいて、PLAの構成は、ANDプレーンをプリバイト用(構成部分8)と、オペコード用(構成部分10)の2種類もち、ORプレーン9,11を共有した形をとる。

これは、先の命令形式でも記述した様に、プリ

バイト部は、アドレッシングモードの情報を有し、オペコード部が命令のオペレーション内容を含むため、ＰＬＡ上でも機能的に、分類することでデコードの容易化と冗長性を排除し、機能別(プリバイトかオペコード)で最小のＰＬＡ(特に、ＡＮＤプレーン)を実現させている。

そして、この２分割されたＰＬＡのＡＮＤプレーン８，１０は、インタラプト制御２１からの入力信号２４により、ＡＮＤプレーンの両方を動作状態にするか、一方ＡＮＤプレーン１０を非動作状態にせしめることもできる。ここで割り込みのシーケンスの制御コードは、全て、プリバイト側のＡＮＤプレーン８にコードが割り付けられており、割り込みの処理時にオペコード側のＡＮＤプレーン１０は非動作状態にある。

演算部２は、上記の制御信号にしたがって、演算やＣＰＵ外部とのデータのアクセスを行なう。

内部バスとしては、基本的にＭＢ，ＤＢ，ＳＢの３種類８ビットバスを有し、各機能部とのデータのやりとりを行なう。

具体的にはＡＣＵ部は、８ビット毎にＩＮＣ／ＤＥＣという増減機能があり、ＡＢＬ，ＡＢＨ，ＳＢという内部バス(各８ビット)からのデータを"００"h，"０１"h，"０２"hで選択的に増減する。

ＩＮＣ／ＤＥＣで演算された結果は、ＣＡＬＬ，ＣＡＬＨ，ＣＡＬＢのラッチに選択的に格納され、ＡＯＢＬ，ＡＯＢＨ，ＡＯＢＢのアドレス・バッファを介して出力される。

ここで選択的というのは、演算結果が常にラッチされるのではなく、アドレス演算時のみラッチして、データ演算時にはラッチされない場合があることを意味する。

しかし、ＲＬＴ２　３５は、ＩＮＣ／ＤＥＣ：Ｂ演算時は常に結果をラッチするデータラッチである。

ＡＣＵ部には、割り込み発生時に強制的に割り込みベクタを発生するＶＥＣＬ，ＶＥＣＨ，ＶＥＣＢ(ベクタアドレス発生回路)や、ＩＮＣ／ＤＥＣを介さずにＤＢバスデータを直接アドレスとして出力するＢＳも配置されている。

機能としては、上述のプログラミングモデルで示したレジスタ群や、データや実効アドレスの演算を行なう８ビットＡＬＵ１９や、シフト演算を行なう８ビットのシフタ２０、アドレス生成を主に行なうＡＣＵ１３がある。

ＡＬＵ１９は、ＭＢ入力側にＩＣ２７をもち、ＩＣ２７は、ＭＢバスから入力される信号を、スルーするか、反転したり、"００"h等の定数データを発生してＡＬＵ１９での演算を補助する。

さらにＤフラグの機能を実現するための１０進補正回路もＡＬＵ１９は含んでいる。

そして内部バス(ＭＢ)のデータのゼロを検出するＺＤＴ１７や分岐命令での分岐条件成立の有無をＰＳＲの状態から検出するＢＲＤＴ１８もある。

特にアドレス生成を主に行なうＡＣＵ部に関しては、８ビット単位に、機能が分離されそれぞれはキャリーが伝搬する構成となっていて、最大２４ビットのアドレス演算を行なう。ここでは、アドレスの演算のみならずデータの演算も可能である。

本ＣＰＵにおいて、実効アドレスの生成は、特に分岐やディスプレースメント付のアドレッシングにおいてＡＵとＡＣＵの両方を使用して演算しており、ＣＳＢ，ＣＳＨ２５，２６は、その際に使用される。

つまり、ＡＬＵ１９からの演算結果によるキャリーやボローをＡＣＵの演算に反映させるためのキャリーのセレクタとしての機能をＣＳＢ，ＣＳＨ２５、２６が持っている。

尚、ＩＮＣ／ＤＥＣからラッチされた演算結果は、ＳＢ，ＡＢＨ，ＡＢＬのバスを介してＰＣ，ＤＢＲ，ＴＲ，ＡＤＨ，ＡＤＬのレジスタデータを選択的に更新する。

その他の機能としては、ＣＰＵのクロックの制御をつかさどる、クロック発生器２２や、周辺システムにＣＰＵの動作状態を知らせる複数の信号を発生するシステム制御２３がある。

さらに、インストラクション・プレデコーダ３３は、命令コードのプリデコードを行い短縮命令の識別や、プリバイト付でオペコードと不当な組

合せ(以下不当命令と称す)の選別などを行なう。

以下に、本ＣＰＵの演算部２の各機能部について説明を行なう。

○　汎用レジスタ

演算,転送時にデータを提供したり、演算,転送後の結果を格納する第３図及び第４図に示す汎用レジスタ群である。

Ｗ０,Ｗ１については、８ビットずつに分けてＲ０,Ｒ２,Ｒ１,Ｒ３の８ビットレジスタとしても命令で区別して使用することができるので、本ＣＰＵでは１６ビットのみならず、８ビットのデータを扱うことができる。

Ｗ２,Ｗ３は、データアクセスの際のポインタとしてアドレッシングモードで指定すれば使用することもできる。

汎用レジスタ群の各レジスタは、ラッチ(セット、リセットなし)で構成され、内部バスに対し、以下の接続関係を有する。

基本的に、ＭＢバスから入力されデータをラッチし、ＤＢ或いはＭＢのバスにラッチされたデータを出力する。

Ｒ２レジスタのみ、除算命令を実行する際のため、入力にＤＢバスを選択可能とする。

R0(W0L)→MBから入力,　　DB あるいはMBへ出力

R2(W0H)→MBあるいはDBから入力,DBあるいはMBへ出力

R1(W1L)→MBから入力,　　DBあるいはMBへ出力

R3(W1H)→MBから入力,　　DBあるいはMBへ出力

W2L　→MBから入力,　　DBあるいはMBへ出力

W2H　→MBから入力,　　DBあるいはMBへ出力

W3L　→MBから入力,　　DBあるいはMBへ出力

W3H　→MBから入力,　　DBあるいはMBへ出力

○　ＦＰＲ(ファーストページレジスタ)

第３図及び第４図に示したＦＰＲは、前述のファースト・ダイレクト・アドレッシングと呼ぶアドレッシングモードで使用される。

ＦＰＲは、ラッチ(セット、リセットなし)で構成され、内部バスに対し、以下の接続関係を有する。

基本的に、ＭＢバスから入力されデータをラッチし、ＤＢのバスにラッチされたデータを、出力する。

FPR → MBから入力 ,　DBへ出力

○　ＩＣ(ＡＬＵに関する入力制御)

第３図に示したＩＣ２７(８ビット)は、ＭＢバスからＡＬＵ１９に入力されるデータを制御する。機能的には、以下の機能を有する。

1. MBバスデータ　　→ALUに入力

2. MBバスデータの反転→ALUに入力

3. "00"hの定数　　→ALUに入力
(MBバスデータは無視する。)

4. "01"hの定数　　→ALUに入力
(MBバスデータは無視する。)

5. "02"hの定数　　→ALUに入力
(MBバスデータは無視する。)

6. "03"hの定数　　→ALUに入力
(MBバスデータは無視する。)

○　ＡＬＵ(演算論理素子)

第３図に示したＡＬＵ１９(８ビット)は、ＤＢバスデータとＩＣからの各８ビットの入力により演算を実施する。

機能的には、ＡＮＤ(論理積)、ＯＲ(論理和)、ＥＸＯＲ(排他的論理和)、ＳＵＭ(加算)がある。

また、ＰＳＲ中のＤフラグの設定により(Ｄ－１ならば)、加算及び減算を同一演算サイクル内で１０進補正する回路も含む。

さらに、ＳＵＭの結果、キャリー・ボロー発生、オーバーフローが発生の検出及びキャリー・ボロー、オーバーフローをラッチする機能も具備する。

特に、キャリー結果は、ＡＬＵ１９が次のＳＵＭを実行するまで保持されるものとする。(ＡＮＤ,ＯＲ,ＥＸＯＲでは変化しない)

○　ＡＬＵシフタ(演算論理素子シフタ)

第３図に示した、ＡＬＵシフタ２８は、８ビットデータの１ビットシフトライトを実施するシフトレジスタで、主に乗算命令で使用される。

このシフトレジスタに入力されるデータは、ＡＬＵ１９のＳＵＭ(加算)の結果であり、最上位ビットには、そのＳＵＭの結果で発生したキャリーが

入力され、シフトの結果最下位より送出される1ビットデータは、ALU19のキャリーとして最終的に保持される。

○ RLT(ALU 結果ラッチ)

第3図に示した、RLT29は、ALU19の演算結果を保持する8ビットラッチである。内部バスに対しては以下の接続関係を有する。

RLT → DBあるいはMBへ出力

ただし、RLT29のデータは、次のALU演算が実行されるまで内容は更新されない。

○ シフタ

第3図に示した、シフタ20はフリップ・フロップで構成され、データの1ビットシフトレフト、シフトライト、ノーシフトのいずれかを制御信号により選択的に実施する。

内部バスに対しては以下の接続関係を有する。

シフタ ← MBから入力, MBへ出力

○ ZDT(ゼロ検出回路)

第3図に示したZDT17は、MBバスの状態をモニタし、MBバスが全ビット"00"hならば、"00"hの検出をしたことを示す信号を発生するゼロ検出回路である。

特に、この信号はPSRレジスタ30中のZフラグに作用し、ALU19等の演算結果がRLT29より、MBバスに出力される時、結果の"00"hを検出してZフラグを"1"にセットする動作を促すために用いられる。

○ PSR(プロセッサ・ステイタス・レジスタ)

第3図に示した、PSR30は、ラッチで構成され内部バスに対しては以下の接続関係を有する。

PSR ← MBから入力 . DBへ出力

機能としては、概要でも記述した様にPSRレジスタ30は現在のCPUの動作状態を示す。

○ BRDT(分岐検出回路)

第3図に示した、BRDT18は、PSR30に接続されており、分岐命令が発生した場合、PSR30の内容から分岐するか否かを判断する信号を発生する。

○ AOBB,AOBH,AOBL(アドレス・出力バッファ)

第3図ACU部16に示した、AOBB,AOBH,AOBLはアドレス出力用のバッファであり、各8ビットで計24ビット(BA7～BA0,A15～A0)のアドレスを出力する。

アドレス出力は、BEのローでハイ・インピーダンス状態になる。

○ VECB,VECH,VECL(ベクタアドレス発生器)

第3図ACU部16に示した、VECB,VECH,VECLは割り込み処理において、ベクタアドレス(24ビット)を発生する。

○ CALB,CALH,CALL(アドレス計算ラッチ)

第3図ACU部16に示した、CALB,CALH,CALLはINC/DEC:B:H:Lの演算の結果を選択的に格納するラッチであり、アドレス演算時のみラッチされる。

○ RLT2(結果ラッチ2)

第3図ACU部16に示した、RLT2はINC/DEC:Bの演算の結果を常に格納するラッチである。

○ INC/DEC:B:H:L(インクリメント/デクリメント・ユニット)

第3図ACU部16に示したINC/DEC:B:H:Lはデータの増減を行なう。

各機能部は、8ビット単位で構成され、演算結果で発生したキャリーは、それぞれの上位アドレス増減部(INC/DEC:LならINC/DEC:Hへ,INC/DEC:HならINC/DEC:Bへ)に伝搬され、結局24ビットのアドレス生成を実現することになる。

但し、このINC/DEC:B:H:Lにデータ(各8ビット)は、SB,ABH,ABLのデータバス(各8ビット)を介して入力される。

各INC/DEC:B:H:Lは、このデータについて基本的に次の動作を選択的に行なう。

1．現状データの保持

2．"01"hのインクリメントあるいはデクリメント。

3．"02"hのインクリメントあるいはデクリ

メント。

○ BS(バス セレクト)

第3図ACU部16に示した、BSは実効アドレスを発生する際、CPU外部から入力されたデータ(8ビット)をINC/DEC:Lを介することなく、DBバスから、直接AOBLに入力するためのデータの選択の機能を有する。

前記のファースト・ダイレクト・アドレッシングのような場合、実効アドレスのためのオペランドデータ(8ビット)をフェッチするサイクルの次に、すぐに実効アドレスを出力しなければならないが、この場合、INC/DEC:Lを介すれば遅延が生じる。

そこでこのBSを用いて、オペランドデータ(DIL)をDBバスに乗せ、BSで選択することにより、高速にAOBLを書き換えることができる。

○ CSB,CSH(キャリーセレクタ)

第3図ACU部16に示した、CSB,CSH25、26はデータの演算時に、INC/DEC:B,INC/DEC:Hに入力されるキャリーが

めて演算ができる。

一方、通常のプログラムカウンタのインクリメント動作の場合には、ACU16のみを用いて、ALU19は別のオペレーションのための演算を行なうことができる。

この時ALU19のキャリーは無視されACUL16から発生したキャリーがCSH25を介してACUHに入力される。

○ PBC,PCH,PCL(プログラム・カウンタ)

24ビットのプログラム・カウンタ・レジスタである。

このレジスタのインクリメントは、INC/DEC:B:H:Lを用いて行なう。

内部バスに対しては以下の接続関係を有する。

PBD → SBから入力，DBあるいはSBへ出力

PCH → ABHから入力，MBあるいはABHへ出力

PCL → ABLから入力，DBあるいはABLへ出力

○ TR,ADH,ADL(テンポラリ・レジスタ)

INC/DECの下位側(INC/DEC:HならINC/DEC:L,INC/DEC:BならINC/DEC:H)からか、あるいはALU19で発生されたキャリーにするかを選択する機能を有する。

従って、このCPUでは実効アドレス発生の際のディスプレースメントデータの加算や、プログラム相対アドレスで分岐の際にアドレスの計算を行なうことは、ALU19とACU16を共用して行なう。

例えば、24ビットデータに8ビットのディスプレースメントを加算して、実効アドレスを発生するアドレッシングの場合、24ビットデータ中のビット7～ビット0とディスプレースメントデータ(8ビット)の加算をALU19で行ない、24ビットの残り(ビット23～ビット16)をACU部16で演算する。

ALU19で加算の結果キャリーが発生した場合、このキャリーは、CSH25を介して、ACUHに入力され、ACU16はこの桁上がりを含

各8ビットのテンポラリ・データラッチである。CPU外部からは見えない。演算結果を一時的に格納する。

TR → DBあるいはSBから入力，SBへ出力

ADH → MBあるいはABHから入力，ABHへ出力

ADL → DBあるいはABLから入力，ABLへ出力

○ SPH,SPL(スタック・ポインタ・レジスタ)

16ビットのスタック・ポインタ・レジスタである。

内部バスに対しては以下の接続関係を有する。

SPH →MBから入力 ，MBあるいはDBへ出力

SPL →MBから入力 ，DBへ出力

○ DBR(データ・バンク・レジスタ)

8ビットのバンク・レジスタである。基本的にデータアクセスの際のバンクアドレスは、このレジスタ値が出力される。但し、PSR中にモード・フラグ(M1,M0)の状態により、バンクアドレス値は変動する。

また、DBRは、SBバスを介して入力されて

おり、ＤＢＲ値の増減にも任意に対応できる。

内部バスに対しては以下の接続関係を有する。

DBR → MBあるいはSBから入力，DBあるいはSBへ出力

○ ＤＩＬ(データ入力ラッチ)

８ビットのラッチである。外部データは、このラッチに入力される。

ＤＩＬ１５は、制御部１に対しては命令コードを供給し、演算部２には、内部バス(ＤＢ，ＭＢ，ＳＢ)に対しデータを供給する。

ＣＰＵ内部に対しては以下の接続関係を有する。

DIL → D7～D0から入力，DB,MB,SBあるいは制御部へ出力

○ ＤＯＬ(データ出力ラッチ)

８ビットのラッチである。外部に出力されるデータは、このラッチに入力される。

ＣＰＵ内部に対しては以下の接続関係を有する。

DIL → DBあるいはMBから入力，D7～D0へ出力

以下に、本ＣＰＵの制御部１の各機能部について説明を行なう。

○ システム制御

ＣＰＵの動作状態を知らせるための複数の信号を発生する。

ＢＳＶＴ－－－プロセッサ動作状態出力
（ベクタアドレス出力中を示す）

ＢＳＤＡ－－－プロセッサ動作状態出力
（データアクセスを示す）

ＢＳＰＡ－－－プロセッサ動作状態出力
（プログラムアクセスを示す）

ＢＳＯＦ－－－プログラム動作状態出力
（命令フェッチを示す）

ＢＳＭＬ－－－プロセッサ動作状態出力
（メモリロック状態を示す）

ＲＷＢ，ＲＢ，ＷＢ－－－リードライト状態出力

ＢＥ－－－バスイネーブル入力

○ インタラプト制御

ＣＰＵの割り込みを制御する。

ＲＥＳ　－－－リセット割り込み入力

ＮＭＩ　－－－ノンマスカブル割り込み入力

ＩＲＯ　－－－割り込み入力

○ インストラクション・プレ・デコーダ

基本的に次の３つの機能部を有する。

１．ＰＬＡでのデコードではタイミング的に間に合わない場合、プレデコーダで予めデコードして制御信号を発生する。

[１サイクル命令の検出，外部制御信号の発生制御，ＴＣＵ７の制御等]

２．ＰＬＡコードの最小化のためデコードを補助する。

[短縮命令の検出，命令で扱うデータサイズの検出等]

３．不当命令，ソフトウェアインターラプト命令の検出。

○ クロック発生器

ＣＰＵ内部用のクロックの発生。あるいは、外部システム用システム・クロックを発生する。

ＷＡＩＴ－－－プロセッサ停止入力

ＬＳＰ　－－－バスサイクル変更用入力

ＣＬＫ　－－－ＣＰＵクロック入力

Ｓ１，Ｓ２－－－システム・クロック出力

ＩＳＥ０～３－－－割り込み(ＩＲＯ)選択入力

ＷＡＫＥ　－－－プロセッサ停止命令の解除入力

○ ＴＣＵ(タイミング制御ユニット)

命令実行の動作シーケンスを制御する。

○ ＥＣＩ(イクスキュージョン制御インタフェース)

ＰＬＡの命令デコード結果を受け演算部２にタイミングを整えた制御信号を発生する機能を有す。

○ オペコードＩＲ(バッファ)，
プリバイトＩＲ(インストラクション・レジスタ)

命令を格納するインストラクションレジスタ。

○ プリバイト　ＡＮＤプレーン，オペコード
ＡＮＤプレーン，ＯＲプレーン

ＡＮＤ－ＯＲで構成された命令デコード用ＰＬＡ。

以上概説したようなＣＰＵにおいて、実行アドレスを計算する構成部分のみを第３図より抜き出し第１図に示す。尚、第１図に示す構成部分の内、

第３図に構成部分に相当するものについては同じ符号を付している。

本実施例のＣＰＵは各メモリの実行アドレス計算をデータの演算実行用のＡＬＵを用いて、かつ増減器を用いて行うが、実行アドレスを計算する際、実行アドレスを構成する例えば２４ビットのすべてを計算する必要はなく、本ＣＰＵでは、実行アドレスを生成する加算数と被加算数とのデータを８ビットずつ３分割し、ＡＬＵにてそれぞれの下位側よりアドレスを計算し下位側からの桁上がりが発生した場合にはそのキャリーを上位側の増減器に送出し加減算を行い実行アドレスを生成する。

第１図において、レジスタ群５０は、第３図に示すＦＰＲ、Ｒ０、Ｗ２Ｌ、等の汎用レジスタであり、このレジスタ群５０の出力側は、Ａ増減器５１、Ｂ増減器５２、Ｃ増減器５３に接続されるとともに、ＡＬＵ１９に接続される。

Ａ増減器５１の出力側は、ＡＬＵ１９より供給される信号とＡ増減器５１より供給される信号とのいずれかを選択するキャリー選択Ａ回路２５を介してＢ増減器５２に接続される。Ｂ増減器５２の出力側は、Ｂ増減器５２より供給される信号あるいはＡＬＵ１９より供給される信号のいずれかを選択するキャリー選択Ｂ回路２６を介してＣ増減器５３に接続される。

尚、Ａ増減器５１は、第３図に示すＩＮＣ/ＤＥＣ:Ｌに相当し、Ｂ増減器５２はＩＮＣ/ＤＥＣ:Ｈに相当し、Ｃ増減器５３はＩＮＣ/ＤＥＣ:Ｂに相当する。又、キャリー選択Ａ回路２５は第３図に示すＣＳＨにであり、キャリー選択Ｂ回路２６はＣＳＢである。

このように構成される本実施例のＣＰＵにおける動作を以下に説明する。

本ＣＰＵの概略にて説明したように、本ＣＰＵが有する３つのアドレッシングのアドレスの発生方法を第２図aないし第２図cに示す。

レジスタ・インダイレクト・８ビットまたは１６ビットディスプレースメントでは、レジスタ群５０内の複数のレジスタより読み出したＤＢＲデータとＷ２あるいはＷ３レジスタのデータとに、オペランドデータとして８ビットあるいは１６ビットのディスプレースメントを符号拡張したアドレスが実行アドレスとなる。

ロングダイレクト・１６ビットレジスタディスプレースメントでは、オペランドデータとしてプログラムから読んだ２４ビットのデータに、レジスタＷ２あるいはＷ３のディスプレースメントデータを符号拡張して加えたアドレスが実行アドレスとなる。

まず、レジスタ・インダイレクト・８ビットディスプレースメントでは、８ビットのディスプレースメントとレジスタＷ２のデータの下位８ビットをＡＬＵ１９で計算し、その結果におけるキャリー信号がキャリー選択Ａ回路２５に送出され、キャリー選択Ａ回路２５は、ＡＬＵ１９からのキャリー信号をＢ増減器５２へ送出する。一方、Ｂ増減器５２及びＣ増減器５３には、それぞれレジスタＷ２の上位データ、ＤＢＲのデータが転送され、Ａ、Ｂ、Ｃの各増減器５１、５２、５３は、符号拡張の値より定数０の増加か定数０の減少かを選択して動作を実行する。このとき、キャリー選択Ｂ回路２６は、Ｂ増減器５２からのキャリー信号を選択する。そして、Ｂ増減器５２、Ｃ増減器５３の増減結果とＡＬＵ１９の結果が、計算された実行アドレスとなる。さらに、１６ビットデータの場合には、もう１バイトアクセスする必要があるので、上記の計算されたアドレスを増減器で定数１の増加をさせて読むアドレスを得る。

次に、レジスタ・インダイレクト・１６ビットディスプレースメントでは、１６ビットディスプレースメントの下位８ビットとレジスタＷ２の下位８ビットをＡＬＵ１９にて計算し、その結果をＡ増減器５１に送出し、さらにそのキャリー信号を使用して１６ビットのディスプレースメントの上位８ビットとレジスタＷ２の上位８ビットをＡＬＵ１９にて計算する。その結果のキャリー信号がキャリー選択Ｂ回路２６に供給され、キャリー選択Ｂ回路２６はＡＬＵ１９からのキャリー信号をＣ増減器５３に送出する。一方、Ｃ増減器５３には、

ＤＢＲのデータが転送され、増減器全体は符号拡張の値により定数０の増加か定数０の減少かを選択して動作を実行する。このとき、キャリー選択Ａ回路２５は、Ａ増減器５１からのキャリー信号を選択する。そしてＣ増減器５３の増減結果、ＡＬＵ１９の結果、Ａ増減器５１の増減結果が計算された実行アドレスとなる。

次に、ロングダイレクト・１６ビットレジスタ・ディスプレースメントの場合は、オペランドとレジスタの関係がレジスタ・インダイレクト・１６ビット・ディスプレースメントと入れ代わるだけであり、動作は同じである。

このようにＡＬＵ１９とＡ、Ｂ、Ｃの各増減器５１、５２、５３にてアドレスを計算することで、ＡＬＵ単体で計算する場合に比べて演算の並列性を失わず、演算処理サイクルを短くすることができる。

又、ＡＬＵ及び増減器で計算することにより、データ演算用のＡＬＵの他にアドレス計算専用のフルアダーを有する場合に比べてハードウエアを小さくすることができる。

【発明の効果】

以上詳述したように本発明によれば、演算処理を行うＡＬＵにて演算処理の結果発生するキャリー信号を増加器にて被加算数に加算することでアドレスデータを得るようにしたことより、演算の並列性を失わず、処理サイクルを短くすることができ、又、アドレスデータ計算専用のＡＬＵが不要であるからＣＰＵのハードウエアを小さくすることができる。

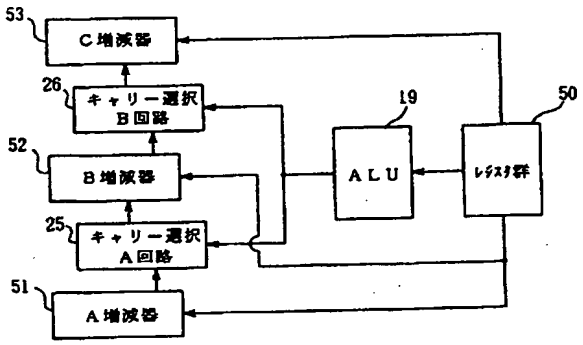
４．図面の簡単な説明

第１図は本発明のＣＰＵの構成部分の内、アドレス計算にかかる構成部分を示したブロック図、第２図aないし第２図cは本発明のＣＰＵが有するアドレッシングのアドレスの発生方法を示す図、第３図は本発明のＣＰＵの全体構成を示すブロック図、第４図は本発明のＣＰＵのプログラミングモデル、第５図aないし第５図cは本発明のＣＰＵの命令形式について示した図である。

１９…ＡＬＵ、２５…キャリー選択Ａ回路、２６…キャリー選択Ｂ回路、５１…Ａ増減器、５２…Ｂ増減器、５３…Ｃ増減器。

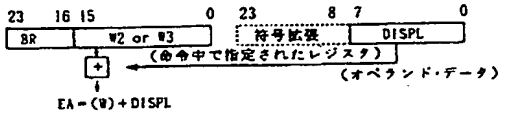
特許出願人　株式会社リコー
代理人　弁理士　青山葆　外１名

第１図



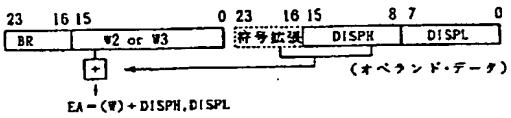
第 2 図 a

レジスタ・インダイレクト・8ビットディスプレースメント



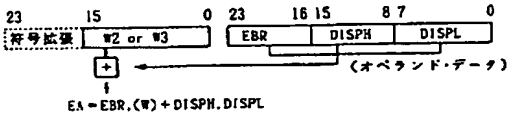
第 2 図 b

レジスタ・インダイレクト・16ビットディスプレースメント



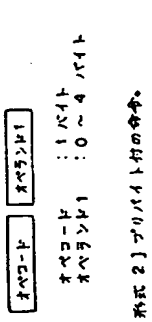
第 2 図 c

ロング・ダイレクト・16ビットレジスタディスプレースメント



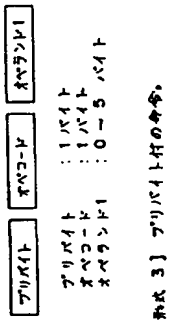
第 5 図 a

[形式 1] プリバイトなしの命令



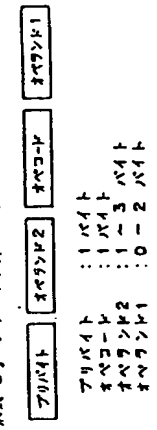
第 5 図 b

[形式 2] プリバイト付の命令。

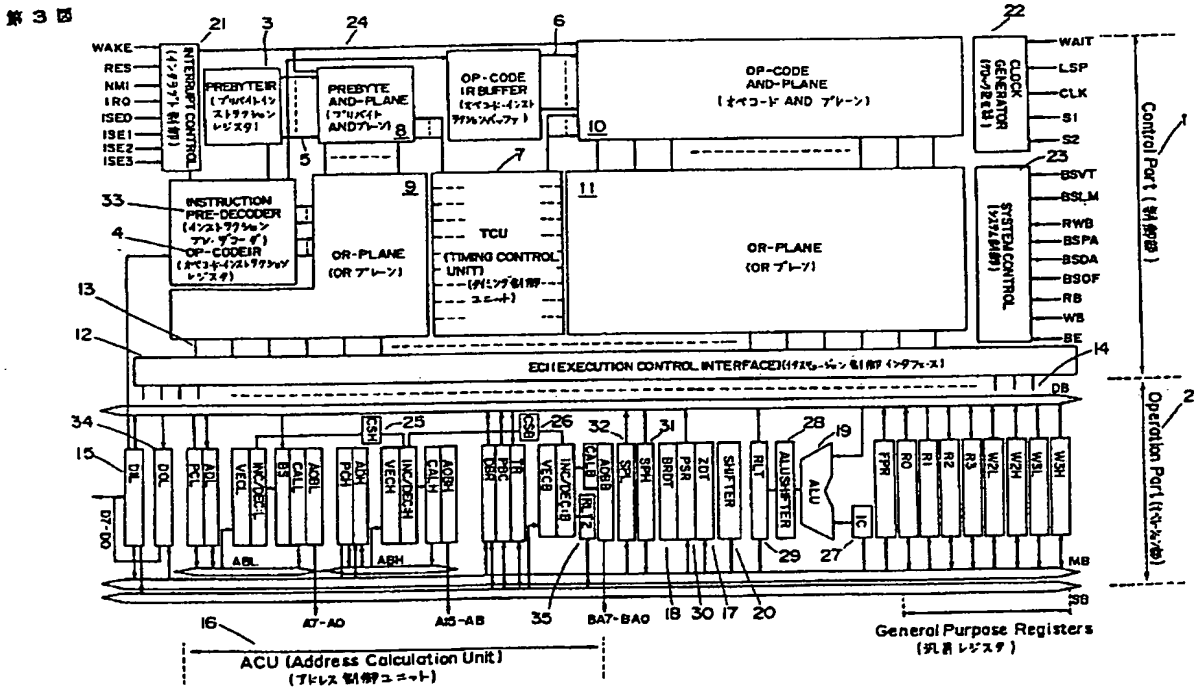


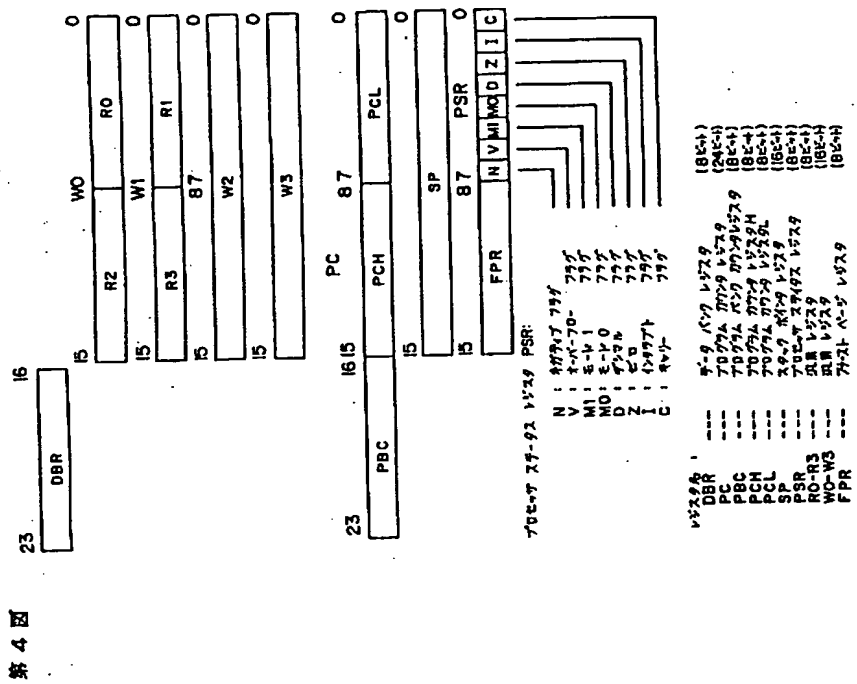
第 5 図 c

[形式 3] プリバイト付の命令。



第 3 図





第4図

【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第６部門第３区分
【発行日】平成１０年（１９９８）９月１１日

【公開番号】特開平３－１５２６２４
【公開日】平成３年（１９９１）６月２８日
【年通号数】公開特許公報３－１５２７
【出願番号】特願平１－２９１７９９
【国際特許分類第６版】
G06F 9/34
7/50

【ＦＩ】
G06F 9/36 310
7/50 E
Z
9/36 330 B

手続補正書

平成 8年10月25日

特許庁長官殿

1. 事件の表示
平成０１年特許願第２９１７９９号

2. 補正をする者
事件との関係 特許出願人
名称 株式会社リコー

3. 代理人
住所 〒540
大阪府大阪市中央区城見１丁目３番７号 ＩＭＰビル
青山特許事務所
電話(06)949-1261
FAX (06)949-0361
氏名 弁理士 (6214) 青山 葆

4. 補正命令の日付
自発（出願審査請求と同時）

5. 補正の対象
明細書：「発明の詳細な説明」の欄

6. 補正の内容
明細書中、次の箇所を補正します。
発明の詳細な説明の欄
(1) 第１４頁第３行に「ＡＵ」とあるを「ＡＬＵ」と補正する。
(2) 第１４頁第１８行に「プレ」とあるを「プリ」と補正する。
(3) 第１４行第１９行に「行い」とあるを「行ない」と補正する。
(4) 第２１頁第２０行から第２２頁第１行にかけて「ラッチ」とあるを、「フリップフロップ」と補正する。
(5) 第２５頁第１１行、同頁第１３行、第２６頁第９行、及び同頁第１５行に「レジスタ」とあるを「ラッチ」と補正する。
(6) 第２５頁第１６行に「ＰＢＤ」とあるを「ＰＢＣ」と補正する。
(7) 第２９頁第２０行、及び第３０頁第１行に「ＩＲＯ」とあるを「ＩＲＱ」と補正する。

以上